

Claims

1. A method for network communications, the method comprising:
 - establishing a first connection between a client and a first protocol service using a first protocol; and
 - communicating between the client and the first protocol service via a plurality of secondary protocols encapsulated within the first protocol,
 - wherein at least one of the secondary protocols comprises a plurality of virtual channels.
2. The method of claim 1 further comprising:
 - establishing a second connection between the first protocol service and a host service using one of the secondary protocols; and
 - communicating between the first protocol service and the host service via one of the secondary protocols.
3. The method of claim 1 further comprising:
 - establishing a plurality of second connections between the first protocol service and a plurality of host services using the plurality of secondary protocols, each of the plurality of second connections being established between the first protocol service and a different host service, and each of the plurality of second connections being established using one of the plurality of secondary protocols; and
 - communicating between the first protocol service and the plurality of host services over each of the plurality of second connections via one of the plurality of secondary protocols.
4. The method of claim 1 wherein establishing the first connection between the client and the first protocol service comprises establishing the first connection through an intermediary node.
5. The method of claim 1 wherein the first protocol is communicated over TCP/IP.
6. The method of claim 1 wherein each virtual channel comprises a plurality of protocol packets enabling remote access functionality.
7. The method of claim 1 further comprising compressing the communications at the level of the first protocol.
8. The method of claim 1 further comprising encrypting the communications at the level of the first protocol.

9. The method of claim 1 wherein the first connection is secure and further comprising:
 - establishing a second connection between the first protocol service and a first host service;
 - communicating between the client and the first host service via the first connection and the second connection;
 - interrupting the second connection;
 - establishing a third connection between the first protocol service and a second host service without interrupting the first connection; and
 - communicating between the client and the second host service via the first connection and the third connection.
10. The method of claim 1 wherein one of the secondary protocols are selected from the set of protocols consisting of HTTP, RDP, ICA, FTP, Oscar, and Telnet.
11. A method for providing a client with a reliable connection to a host service, the method comprising:
 - establishing a first connection between the client and a first protocol service using a first protocol, the first protocol for encapsulating a plurality of secondary protocols;
 - establishing a second connection between the first protocol service and the host service using one of the plurality of secondary protocols;
 - maintaining a queue of data packets most recently transmitted via the first connection on at least one of the client and the first protocol service; and
 - upon failure of the first connection:
 - maintaining the second connection;
 - continuing to maintain the queue of data packets most recently transmitted via the first connection; and
 - establishing a third connection between the client and the first protocol service using the first protocol.
12. The method of claim 11 further comprising transmitting at least one of the queued data packets via the third connection.
13. A method for re-connecting a client to a host service, the method comprising:

providing a first connection between the client and an intermediary node, a second connection between the intermediary node and a first protocol service, and a third connection between the first protocol service and the host service;

detecting a disruption in at least one of the first connection and the second connection;

re-establishing the first connection between the client and the intermediary node while maintaining the third connection between the first protocol service and the host service;

receiving at the intermediary node a first ticket and a second ticket;

validating the first ticket;

re-establishing, after the first ticket is validated, the second connection between the intermediary node and the first protocol service;

validating the second ticket; and

linking, after the second ticket is validated, the re-established second connection to the maintained third connection.

14. The method of claim 13 further comprising interrupting, after the disruption in at least one of the first connection and the second connection is detected, any remaining connections of the first connection and the second connection.

15. The method of claim 13 wherein validating the first ticket comprises validating the first ticket using a ticket authority.

16. The method of claim 15 wherein validating the first ticket further comprises transmitting the first ticket from the intermediary node to the ticket authority.

17. The method of claim 13 further comprising receiving at the intermediary node, after the first ticket is validated, an address for the first protocol service.

18. The method of claim 13 further comprising deleting, after the first ticket is validated, the first ticket.

19. The method of claim 18 further comprising generating, after the first ticket is deleted, a replacement first ticket.

20. The method of claim 13 wherein validating the second ticket comprises validating the second ticket using the first protocol service.

21. The method of claim 20 wherein validating the second ticket further comprises transmitting the second ticket from the intermediary node to the first protocol service.
22. The method of claim 13 further comprising deleting, after the second ticket is validated, the second ticket.
23. The method of claim 22 further comprising generating, after the second ticket is deleted, a replacement second ticket.
24. The method of claim 13 further comprising transmitting, from the intermediary node to a ticket authority, a request for the first ticket.
25. The method of claim 13 further comprising generating the first ticket at a ticket authority.
26. The method of claim 25 wherein generating the first ticket comprises generating a random number.
27. The method of claim 25 further comprising generating a handle at the ticket authority.
28. The method of claim 27 further comprising saving, at the ticket authority, a copy of the first ticket, a copy of the handle, and an address for the first protocol service.
29. The method of claim 27 further comprising transmitting the first ticket and the handle from the ticket authority to the intermediary node.
30. The method of claim 29 further comprising using the handle to delete a copy of the first ticket saved at the ticket authority.
31. The method of claim 29 further comprising transmitting the first ticket from the intermediary node to the client.
32. The method of claim 13 further comprising generating the second ticket at the first protocol service.
33. The method of claim 32 wherein generating the second ticket comprises generating a random number.
34. The method of claim 32 further comprising saving, at the first protocol service, a copy of the second ticket and a session number.

35. The method of claim 32 further comprising transmitting the second ticket from the first protocol service to the client.

36. The method of claim 13 wherein at least one of the first ticket and the second ticket is automatically deleted after a pre-determined period of time.

37. A method for re-connecting a client to a host service, the method comprising:

providing a first connection between the client and a first intermediary node, a second connection between the first intermediary node and a first protocol service, and a third connection between the first protocol service and the host service;

detecting a disruption in at least one of the first connection and the second connection;

establishing a fourth connection between the client and a second intermediary node, different from the first intermediary node, while maintaining the third connection between the first protocol service and the host service;

receiving at the second intermediary node a first ticket and a second ticket;

validating the first ticket;

establishing, after the first ticket is validated, a fifth connection between the second intermediary node and the first protocol service;

validating the second ticket; and

linking, after the second ticket is validated, the established fifth connection to the maintained third connection.

38. A method for re-connecting a client to a host service, the method comprising:

providing a first connection between the client and a first protocol service, and a second connection between the first protocol service and the host service;

detecting a disruption in the first connection;

re-establishing the first connection between the client and the first protocol service while maintaining the second connection between the first protocol service and the host service;

receiving at the first protocol service a ticket;

validating the ticket; and

linking, after the ticket is validated, the re-established first connection to the maintained second connection.

39. The method of claim 38 further comprising deleting, after the ticket is validated, the ticket.

40. The method of claim 39 further comprising generating, after the ticket is deleted, a replacement ticket.
41. The method of claim of claim 38 further comprising generating the ticket at the first protocol service.
42. The method of claim 41 wherein generating the ticket comprises generating a random number.
43. The method of claim 41 further comprising saving, at the first protocol service, a copy of the ticket and a session number.
44. The method of claim 41 further comprising transmitting the ticket from the first protocol service to the client.
45. The method of claim 38 wherein the ticket is automatically deleted after a pre-determined period of time.
46. A system for network communications, the system comprising:
 - a first protocol service configured to accept a first connection with a client and communicate with the client via a plurality of secondary protocols encapsulated within a first protocol, wherein at least one of the secondary protocols comprises a plurality of virtual channels.
47. The system of claim 46 wherein the first protocol service is further configured to establish a second connection with a host service and communicate with the host service via one of the secondary protocols.
48. The system of claim 46 wherein the first protocol service is further configured to establish a plurality of second connections with a plurality of host services using the plurality of secondary protocols, each of the plurality of second connections being established with a different host service and each of the plurality of second connections being established using one of the plurality of secondary protocols, and wherein the first protocol service is further configured to communicate with the plurality of host services over each of the plurality of second connections via one of the plurality of secondary protocols.
49. The method of claim 46 wherein the first connection with the client is routed through an intermediary node.

50. The system of claim 46 wherein the first protocol is communicated over TCP/IP.
51. The system of claim 46 wherein each virtual channel comprises a plurality of protocol packets enabling remote access functionality.
52. The system of claim 46 wherein the first protocol service is further configured to compress the communications at the level of the first protocol.
53. The system of claim 46 wherein the first protocol service is further configured to encrypt the communications at the level of the first protocol.
54. The system of claim 46 wherein the first connection is secure and the first protocol service is further configured to establish a second connection with a first host service, interrupt the second connection, and establish a third connection with a second host service without interrupting the first connection.
55. The system of claim 46 wherein one of the secondary protocols are selected from the set of protocols consisting of HTTP, RDP, ICA, FTP, Oscar, and Telnet.
56. A system for providing a client with a reliable connection to a host service, the system comprising:
 - a first protocol service configured to accept a first connection with the client, establish a second connection with the host service, and, upon failure of the first connection: maintain the second connection and accept a third connection from the client; and
 - the host service configured to accept the second connection with the first protocol service and, upon failure of the first connection: maintain the second connection,
 - wherein the first connection and the third connection are each established using a first protocol, the first protocol for encapsulating a plurality of secondary protocols, and wherein at least one of the client and the first protocol service is further configured to maintain, before and upon failure of the first connection, a queue of data packets most recently transmitted via the first connection.
57. The system of claim 56 wherein the client is further configured to transmit at least one of the queued data packets via the third connection.
58. The system of claim 56 wherein the first protocol service is further configured to transmit at least one of the queued data packets via the third connection.

59. A system for re-connecting a client to a host service, the system comprising:
- the client configured to maintain a first connection with an intermediary node;
 - the intermediary node configured to maintain the first connection with the client and a second connection with a first protocol service; and
 - the first protocol service configured to maintain the second connection with the intermediary node and a third connection with the host service, wherein:
 - a disruption is detected in at least one of the first connection and the second connection;
 - the first connection is re-established between the client and the intermediary node while the third connection between the first protocol service and the host service is maintained;
 - a first ticket and a second ticket are transmitted from the client to the intermediary node;
 - the first ticket is validated;
 - the second connection between the intermediary node and the first protocol service is reestablished after the first ticket is validated;
 - the second ticket is validated; and
 - after the second ticket is validated, the re-established second connection is linked to the maintained third connection.
60. The system of claim 59 wherein, after the disruption is detected in at least one of the first connection and the second connection, any remaining connections of the first connection and the second connection are broken.
61. The system of claim 59 wherein the first ticket is validated using a ticket authority.
62. The system of claim 61 wherein the ticket authority is configured to receive the first ticket from the intermediary node and validate the first ticket.
63. The system of claim 62 wherein the ticket authority is further configured to delete, after the first ticket is validated, the first ticket.
64. The system of claim 63 wherein the ticket authority is further configured to generate, after the first ticket is deleted, a replacement first ticket.
65. The system of claim 59 wherein the intermediary node is further configured to receive, after the first ticket is validated, an address for the first protocol service.

66. The system of claim 59 wherein the second ticket is validated using the first protocol service.
67. The system of claim 66 wherein the first protocol service is further configured to receive the second ticket from the intermediary node and validate the second ticket.
68. The system of claim 67 wherein the first protocol service is further configured to delete, after the second ticket is validated, the second ticket.
69. The system of claim 68 wherein the first protocol service is further configured to generate, after the second ticket is deleted, a replacement second ticket.
70. The system of claim 59 wherein the intermediary node is further configured to transmit a request for the first ticket to a ticket authority.
71. The system of claim 59 further comprising a ticket authority configured to generate the first ticket.
72. The system of claim 71 wherein the first ticket comprises a random number.
73. The system of claim 71 wherein the ticket authority is further configured to generate a handle.
74. The system of claim 73 wherein the first ticket authority is further configured to save a copy of the first ticket, a copy of the handle, and an address for the first protocol service.
75. The system of claim 73 wherein the ticket authority is further configured to transmit the first ticket and the handle to the intermediary node.
76. The system of claim 75 wherein the intermediary node is further configured to use the handle to delete a copy of the first ticket saved at the ticket authority.
77. The system of claim 75 wherein the intermediary node is further configured to transmit the first ticket to the client.
78. The system of claim 59 wherein the first protocol service is further configured to generate the second ticket.
79. The system of claim 78 wherein the second ticket comprises a random number.

80. The system of claim 78 wherein the first protocol service is further configured to save a copy of the second ticket and a session number.
81. The system of claim 78 wherein the first protocol service is further configured to transmit the second ticket to the client.
82. The system of claim 59 wherein at least one of the first ticket and the second ticket is configured for automatic deletion after a pre-determined period of time.
83. A system for re-connecting a client to a host service, the system comprising:
- the client configured to maintain a first connection with a first intermediary node;
 - the first intermediary node configured to maintain the first connection with the client and a second connection with a first protocol service;
 - the first protocol service configured to maintain the second connection with the first intermediary node and a third connection with the host service; and
 - a second intermediary node, different from the first intermediary node, wherein:
 - a disruption is detected in at least one of the first connection and the second connection;
 - a fourth connection is established between the client and the second intermediary node while the third connection between the first protocol service and the host service is maintained;
 - a first ticket and a second ticket are transmitted from the client to the second intermediary node;
 - the first ticket is validated;
 - a fifth connection between the second intermediary node and the first protocol service is established after the first ticket is validated;
 - the second ticket is validated; and
 - after the second ticket is validated, the established fifth connection is linked to the maintained third connection.
84. A system for re-connecting a client to a host service, the system comprising:
- the client configured to maintain a first connection with a first protocol service; and
 - the first protocol service configured to maintain the first connection with the client and a second connection with the host service, wherein:
 - a disruption is detected in the first connection;

the first connection is re-established between the client and the first protocol service while the second connection between the first protocol service and the host service is maintained; a ticket is transmitted from the client to the first protocol service; the ticket is validated; and after the ticket is validated, the re-established first connection is linked to the maintained second connection.

85. The system of claim 84 wherein the first protocol service is further configured to delete, after the ticket is validated, the ticket.
86. The system of claim 85 wherein the first protocol service is further configured to generate, after the ticket is deleted, a replacement ticket.
87. The system of claim 84 wherein the first protocol service is further configured to generate the ticket.
88. The system of claim 87 wherein the ticket comprises a random number.
89. The system of claim 87 wherein the first protocol service is further configured to save a copy of the ticket and a session number.
90. The system of claim 87 wherein the first protocol service is further configured to transmit the ticket to the client.
91. The system of claim 84 wherein the ticket is configured for automatic deletion after a pre-determined period of time.